

**In the Claims:**

This listing of claims will replace all prior versions, and listings, of claims in the application.

**Listing of Claims:**

1. (Currently Amended) A computer implemented method for a primary application to provide an indication of the integrity of a secondary application database including the steps of:
  - i. obtaining a first reference reduced representation by:
    - A. applying a process to obtain first schema metadata representative of a database structure of a database from the secondary application;
    - B. creating the first reference reduced representation of the first obtained schema metadata using an algorithm; and
    - C. storing the first reference reduced representation;
  - ii. during execution of a primary application, applying the process to obtain second schema metadata representative of the database structure from the secondary application;
  - iii. creating a second reduced representation of the second obtained schema metadata using the algorithm;
  - iv. ~~comparing the first reference reduced representation with the second reduced representation to provide an indication of the integrity of the secondary application; and~~  
comparing the stored first reference reduced representation with the created second reduced representation so as to provide an indication of the integrity of the database by determining whether the database structure changed during a time between when the first reference reduced representation and second reduced representation were obtained; and
  - v. controlling execution of the primary application dependent on the indication.
2. CANCELLED.

3. (Previously amended) The method as claimed in claim 2 wherein the first or second schema metadata is selected from the set of tables, columns in tables, datatypes of columns, lengths of columns, custom database data types, foreign keys, constraints, stored procedures, views, triggers, indices, and scheduled jobs.
4. (Original) The method as claimed in claim 2 wherein the algorithm is a hash function.
5. (Original) The method as claimed in claim 3 wherein the hash function is one selected from the set of MD5 and CRC32.
6. (Original) The method as claimed in claim 2 wherein the algorithm is a lossless compression algorithm.
7. (Original) The method as claimed in claim 6 wherein the lossless compression algorithm is one selected from the set of zip, gzip, and bzip2.
8. (Previously amended) The method as claimed in claim 2 wherein the first reference reduced representation is stored by embedding the representation within the primary application.
9. (Previously amended) The method as claimed in claim 2 wherein the first reference reduced representation is stored by embedding the representation within configuration files for the primary application.
10. (Original) The method as claimed in claim 2 wherein step (i) is repeated before steps (ii) to (v) at least one time when an expected change occurs to the schema metadata in the database.
11. (Original) The method as claimed in claim 2 wherein the process includes organizing the extracted schema metadata using a nested and determinable method.
12. (Original) The method as claimed in claim 11 wherein the nested and determinable method is by alphabetical listing of the schema metadata elements.

13. (Original) The method as claimed in claim 11 wherein the nested and determinable method is by default database order of the schema metadata elements.
14. (Original) The method as claimed in claim 11 wherein the nested and determinable method is by creation date order of the schema metadata elements.
15. (Original) The method as claimed in claim 11 wherein the nested and determinable method is by table owner of the schema metadata elements.
16. (Original) The method as claimed in claim 2 wherein the execution of the primary application is controlled by halting execution of the primary application.
17. (Original) The method as claimed in claim 2 wherein the execution of the primary application is controlled by the primary application sending an error message to one selected from the set of a user of the primary application, a manager of the primary application, a manager of the database, and the database.
18. (Original) The method as claimed in claim 2 including the step of:
  - i. requesting a schema stability lock of the database.
19. (Original) The method as claimed in claim 2 wherein the process obtains all available schema metadata.
20. (Original) The method as claimed in claim 2 wherein the process only obtains the schema metadata which would affect the primary application if that schema metadata were to change.
21. (Original) The method as claimed in claim 2 wherein the process utilizes SQL92 standard to obtain the schema metadata from the database.
22. (Original) The method as claimed in claim 2 wherein the process utilizes the database's API to obtain the schema metadata from the database.
23. (Original) The method as claimed in claim 22 wherein the database's API is a Java database API.

24. (Currently amended) A computer system for providing an indication the integrity of one or more databases for a plurality of applications including:
- ~~i. — A processor adapted to execute a plurality of applications, adapted to store a plurality of previously calculated reduced representations of schema metadata representative of the structure of one or more databases, to extract a plurality of schema metadata representative of database structure from one or more databases, to newly calculate a plurality of reduced representations from the plurality of extracted schema metadata, and to compare each of plurality of previously calculated reduced representations with its corresponding newly calculated reduced representation to provide an indication of the integrity of one or more databases; and~~
  - ~~ii. — one or more databases adapted to receive requests for schema metadata from the plurality of applications and to transmit schema metadata to the plurality of applications dependent on said indication.~~
  - i. a plurality of applications adapted to carry out the method of claim 1; and
  - ii. a database adapted to receive requests for schema metadata from the plurality of applications and to transmit schema metadata to the plurality of applications dependent on said indication.
25. (Original) The system as claimed in claim 24 wherein the schema metadata is selected from the set of tables, columns in tables, datatypes of columns, lengths of columns, custom database data types, foreign keys, constraints, stored procedures, views, triggers, indices, and scheduled jobs.
26. (Original) The system as claimed in claim 24 wherein the reduced representations are calculated using a hash function.
27. (Original) The system as claimed in claim 26 wherein the hash function is one selected from the set of MD5 and CRC32.
28. (Original) The system as claimed in claim 24 wherein reduced representations are calculated using a lossless compression algorithm.

29. (Original) The system as claimed in claim 28 wherein the lossless compression algorithm is one selected from the set of zip, gzip, and bzip2.
30. (Original) The system as claimed in claim 24 wherein each previously calculated reduced representation is stored by embedding the representation within its associated application.
31. (Original) The system as claimed in claim 24 wherein each previously calculated reduced representation is stored by embedding the representation within configuration files for its associated application.
32. (Original) The system as claimed in claim 24 wherein each schema metadata is organized using a nested and determinable method before its reduced representation is calculated.
33. (Original) The system as claimed in claim 32 wherein the nested and determinable method is by alphabetical listing of the schema metadata elements.
34. (Original) The system as claimed in claim 32 wherein the nested and determinable method is by default database order of the schema metadata elements.
35. (Original) The system as claimed in claim 32 wherein the nested and determinable method is by creation date order of the schema metadata elements.
36. (Original) The system as claimed in claim 32 wherein the nested and determinable method is by table owner of the schema metadata elements.
37. (Original) The system as claimed in claim 24 wherein the result of each comparison controls execution of its associated application
38. (Original) The system as claimed in claim 37 wherein the execution of the application is controlled by halting execution of the application.
39. (Original) The system as claimed in claim 37 wherein the execution of the application is controlled by the application sending an error message to one

selected from the set of a user of the application, a manager of the application, a manager of the associated database, and the associated database.

40. (Original) The system as claimed in claim 24 wherein the plurality of applications are further adapted to request a schema stability lock of the one or more databases.
41. (Original) The system as claimed in claim 24 wherein each application is adapted to extract all available schema metadata from each database.
42. (Original) The system as claimed in claim 24 wherein each application is adapted to extract the schema metadata which would affect the application if that schema metadata were to change.
43. (Original) The system as claimed in claim 24 wherein each application is adapted to utilize SQL92 standard to extract the schema metadata from each database.
44. (Original) The system as claimed in claim 24 wherein each application is adapted to utilize the database's API to extract the schema metadata from each database.
45. (Original) The system as claimed in claim 44 wherein the database's API is a Java database API.
46. (Currently amended) A system comprising a processor, memory and verification engine for providing an indication of the integrity of a database for an application wherein:
  - i. an application;
  - ~~ii. the memory is adapted to store a reduced representation of schema metadata representative of the structure of a database; and~~
  - ~~iii. the verification engine is adapted upon connection to a database to obtain a reduced representation of schema metadata representative of the structure of the database from the database and compare it with the stored reduced representation in order to provide an indication of the integrity of the database and to control the application based on said indication.~~
  - ii. a stored reduced representation of schema metadata representative of the

- structure of a database; and
- iii. a verification engine which upon connection to a database executes the method of claim 1.
47. (Original) The system as claimed in claim 46 wherein the schema metadata is selected from the set of tables, columns in tables, datatypes of columns, lengths of columns, custom database data types, foreign keys, constraints, stored procedures, views, triggers, indices, and scheduled jobs.
48. (Original) The system as claimed in claim 46 wherein the reduced representations are calculated using a hash function.
49. (Original) The system as claimed in claim 46 wherein the stored reduced representation is stored by embedding the representation within the application.
50. (Original) The system as claimed in claim 48 wherein each schema metadata is organized using a nested and determinable method before its reduced representation is calculated.
51. (Original) The system as claimed in claim 46 wherein the application is controlled by halting execution of the application.
52. (Original) The system as claimed in claim 46 wherein the application is controlled by the application sending an error message to one selected from the set of a user of the application, a manager of the application, a manager of the associated database, and the associated database.
53. (Cancelled)
54. (Cancelled)
55. (Previously Presented) Storage media containing software as claimed in claim 1.